Modelo de regresión para la predicción de precios de automóviles

El problema que resolveremos en estos entregables es la predicción del precio de un automobil. Supongamos que tenemos un sitio web donde la gente puede vender y comprar autos usados. Al publicar un anuncio en nuestro sitio web, los vendedores a menudo luchan por conseguir un precio significativo.

Queremos ayudar a nuestros usuarios con recomendaciones automáticas de precios. Pedimos a los vendedores que especifiquen el modelo, la marca, el año, el kilometraje y otras características importantes del auto, y con esta información, queremos sugerir el mejor precio.

1) Análisis exporatorio de

1.1. Carga y preparación de los datos

Vamos a utilizar el set de datos identificado como `data.csv` , en el cual, se encuentra la información de los vehículos vendido.

```
df = read.csv("data.csv", sep=",", header = TRUE)
head(df)
```

```
##   Make      Model Year            Engine.Fuel.Type Engine.HP
## 1  BMW 1 Series M 2011 premium unleaded (required)       335
## 2  BMW   1 Series 2011 premium unleaded (required)       300
## 3  BMW   1 Series 2011 premium unleaded (required)       300
## 4  BMW   1 Series 2011 premium unleaded (required)       230
## 5  BMW   1 Series 2011 premium unleaded (required)       230
## 6  BMW   1 Series 2012 premium unleaded (required)       230
##   Engine.Cylinders Transmission.Type    Driven_Wheels Number.of.Doors
## 1                6            MANUAL rear wheel drive               2
## 2                6            MANUAL rear wheel drive               2
## 3                6            MANUAL rear wheel drive               2
## 4                6            MANUAL rear wheel drive               2
## 5                6            MANUAL rear wheel drive               2
## 6                6            MANUAL rear wheel drive               2
##                        Market.Category Vehicle.Size Vehicle.Style
## 1 Factory Tuner,Luxury,High-Performance      Compact         Coupe
## 2                    Luxury,Performance      Compact   Convertible
## 3               Luxury,High-Performance      Compact         Coupe
## 4                    Luxury,Performance      Compact         Coupe
## 5                               Luxury      Compact   Convertible
## 6                    Luxury,Performance      Compact         Coupe
##   highway.MPG city.mpg Popularity  MSRP
## 1          26       19       3916 46135
## 2          28       19       3916 40650
## 3          28       20       3916 36350
## 4          28       18       3916 29450
## 5          28       18       3916 34500
## 6          28       18       3916 31200
```

```
dim(df)
```

```
## [1] 11914    16
```

Al inspeccionar los datos notamos que hay algunas inconsistencias en este conjunto de datos: los nombres de las columnas a veces tienen espacios, ya veces tienen guiones bajos (_). Lo mismo ocurre con los valores de las caracterìsticas: a veces están en mayásculas, y a veces son cadenas cortas con espacios. Esto es incómodo y confuso, pero podemos solucionarlo normalizando, es decir,

```
library(janitor)
```

```
## Warning: package 'janitor' was built under R version 3.6.3
```

```
##
## Attaching package: 'janitor'
```

```
## The following objects are masked from 'package:stats':
##
##     chisq.test, fisher.test
```

```
df = clean_names(df)
head(df)
```

```
##   make      model year            engine_fuel_type engine_hp
## 1  BMW 1 Series M 2011 premium unleaded (required)       335
## 2  BMW   1 Series 2011 premium unleaded (required)       300
## 3  BMW   1 Series 2011 premium unleaded (required)       300
## 4  BMW   1 Series 2011 premium unleaded (required)       230
## 5  BMW   1 Series 2011 premium unleaded (required)       230
## 6  BMW   1 Series 2012 premium unleaded (required)       230
##   engine_cylinders transmission_type    driven_wheels number_of_doors
## 1                6            MANUAL rear wheel drive               2
## 2                6            MANUAL rear wheel drive               2
## 3                6            MANUAL rear wheel drive               2
## 4                6            MANUAL rear wheel drive               2
## 5                6            MANUAL rear wheel drive               2
## 6                6            MANUAL rear wheel drive               2
##                         market_category vehicle_size vehicle_style
## 1 Factory Tuner,Luxury,High-Performance      Compact         Coupe
## 2                    Luxury,Performance      Compact   Convertible
## 3               Luxury,High-Performance      Compact         Coupe
## 4                    Luxury,Performance      Compact         Coupe
## 5                                Luxury      Compact   Convertible
## 6                    Luxury,Performance      Compact         Coupe
##   highway_mpg city_mpg popularity  msrp
## 1          26       19       3916 46135
## 2          28       19       3916 40650
## 3          28       20       3916 36350
## 4          28       18       3916 29450
## 5          28       18       3916 34500
## 6          28       18       3916 31200
```

```
df$model <- gsub(" ", "_", df$model)
df$engine_fuel_type <- gsub(" ", "_", df$engine_fuel_type)
df$driven_wheels <- gsub(" ", "_", df$driven_wheels)
head(df)
```

```
##   make      model year            engine_fuel_type engine_hp
## 1  BMW 1_Series_M 2011 premium_unleaded_(required)       335
## 2  BMW   1_Series 2011 premium_unleaded_(required)       300
## 3  BMW   1_Series 2011 premium_unleaded_(required)       300
## 4  BMW   1_Series 2011 premium_unleaded_(required)       230
## 5  BMW   1_Series 2011 premium_unleaded_(required)       230
## 6  BMW   1_Series 2012 premium_unleaded_(required)       230
##   engine_cylinders transmission_type    driven_wheels number_of_doors
## 1                6            MANUAL rear_wheel_drive              2
## 2                6            MANUAL rear_wheel_drive              2
## 3                6            MANUAL rear_wheel_drive              2
## 4                6            MANUAL rear_wheel_drive              2
## 5                6            MANUAL rear_wheel_drive              2
## 6                6            MANUAL rear_wheel_drive              2
##                         market_category vehicle_size vehicle_style
## 1 Factory Tuner,Luxury,High-Performance      Compact         Coupe
## 2                    Luxury,Performance      Compact   Convertible
## 3                Luxury,High-Performance      Compact         Coupe
## 4                    Luxury,Performance      Compact         Coupe
## 5                                Luxury      Compact   Convertible
## 6                    Luxury,Performance      Compact         Coupe
##   highway_mpg city_mpg popularity  msrp
## 1          26       19       3916 46135
## 2          28       19       3916 40650
## 3          28       20       3916 36350
## 4          28       18       3916 29450
## 5          28       18       3916 34500
## 6          28       18       3916 31200
```

1.2 Análisis de variables objetivo

Como vemos, este conjunto de datos contiene mùltiples columnas:

- marca(make): marca del coche (BMW, Toyota, etc.)
- modelo(model): modelo de coche
- año(year): año de fabricación del coche
- tipo_de_combustible_del_motor(engine_fuel_type): tipo de combustible que necesita el motor (diesel, elèctrico, etc.)
- engine_hp : potencia del motor
- engine_cylinders: nùmero de cilindros del motor
- tipo_de_transmisión (transmission_type): tipo de transmisión (automótica o manual)
- ruedas motrices (driven_wheels): delanteras, traseras, todas
- número_de_puertas(number_of_doors): número de puertas que tiene el coche
- categoría_de_mercado(market_category): lujo, crossover, etc.
- tamaño_del_vehículo(vehicle_size): compacto, mediano o grande
- estilo_del_vehículo(vehicle_style): sedan o descapotable

- highway_mpg: millas por galón (mpg) en carretera
- city_mpg: millas por galón en la ciudad
- popularidad (popularity): número de veces que se menciona el coche en una corriente de Twitter
- msrp: precio de venta sugerido por el fabricante

La columna MSRP (precio de venta al público sugerido por el fabricante) es nuestra variable objetivo, es decir, es el valor que queremos aprender a predecir.

Uno de los primeros pasos del análisis exploratorio de datos debera ser siempre mirar còmo son los valores de y (MSRP). Para ello, solemos comprobar la distribución de y(descripción visual de los posibles valores de y) y la frecuencia con la que se producen.

Utilice Seaborn o Matplotlib para trazar un histograma de la varible objetivo.

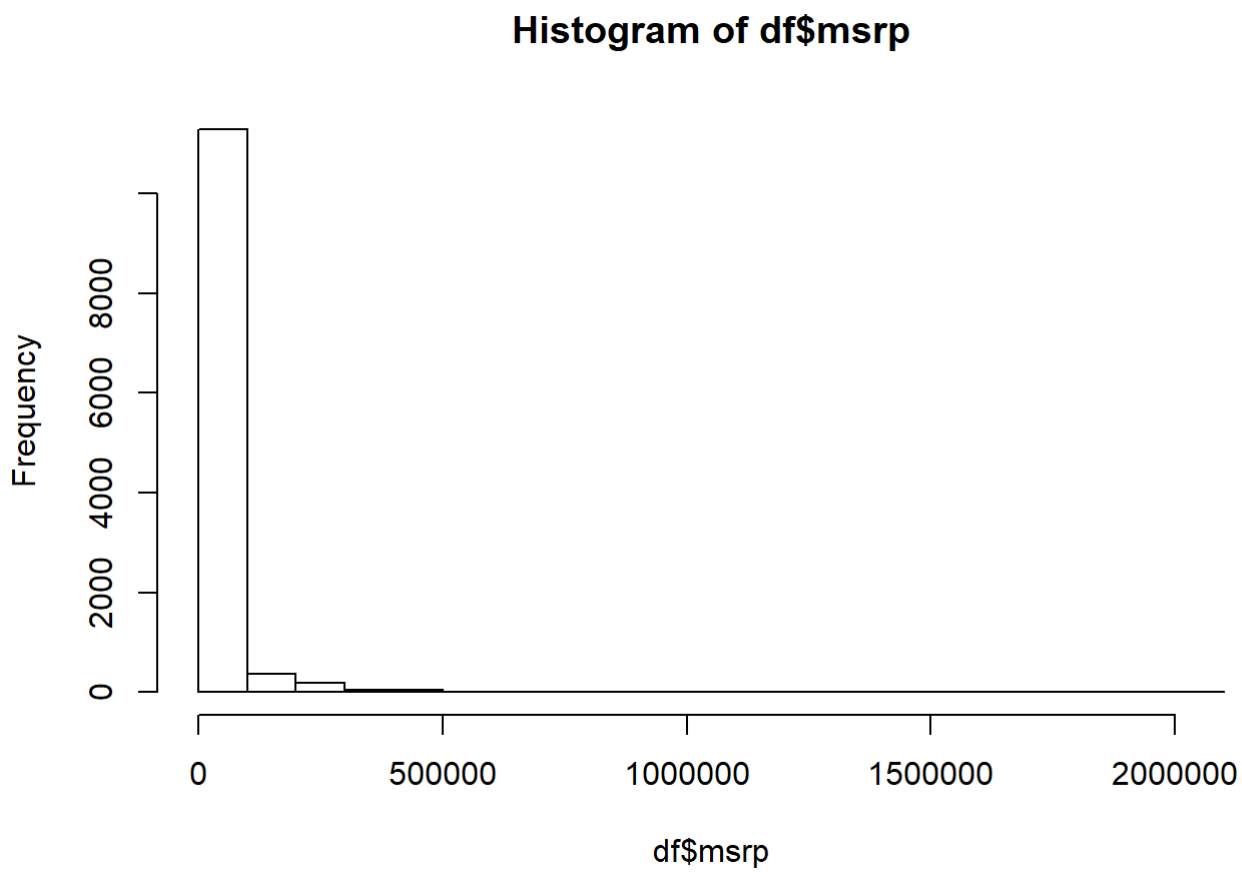
Entrenamiento

Realice un histograma para la varible objetivo.

Luego de graficar los datos, ¿qué observa?. Explique.

¿Es posible mejorar la visualización de los datos?, en caso afirmativo, explique por quÃ©.

```
hist(df$msrp, freq = TRUE)
```



A los datos será necesaria realizarle la siguiente transformación,

[y_{nueva} = (y+1)]

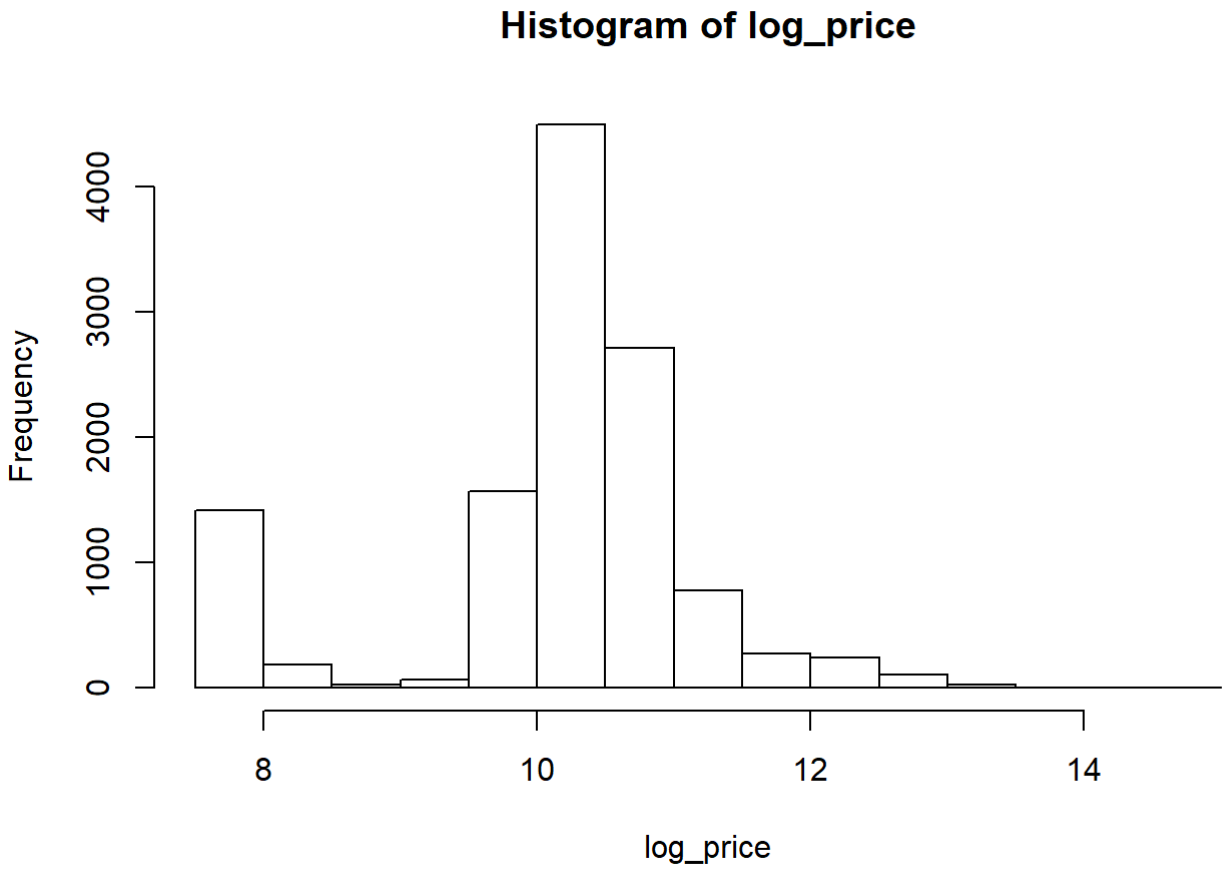
Entrenamiento:

Explique por què es necesaria esta transformación.

Aplique la función `log1p()`

Realice el histograma para los datos transformados y compare con el histograma anterior.¿Qué observa?

```
log_price = log1p(df$msrp)
hist(log_price, freq = TRUE)
```



1.3 Valores faltantes

Comprobemos si hay valores faltantes o perdidos en los datos. Este paso es importante porque, normalmente, los modelos de aprendizaje automático no pueden tratar los valores faltantes automáticamente.

En R lo hacemos de la forma siguiente:

```
sapply(df, function(x) sum(is.na(x)))
```

```
##              make             model              year  engine_fuel_type
##                 0                 0                 0                 0
##         engine_hp  engine_cylinders transmission_type     driven_wheels
##                69                30                 0                 0
##   number_of_doors   market_category      vehicle_size     vehicle_style
##                 6                 0                 0                 0
##       highway_mpg          city_mpg        popularity              msrp
##                 0                 0                 0                 0
```

Entrenamiento:

¿Qué observa de la salida de la función aplicada?

1.4 División de los datos

Tenemos que dividir el conjunto de datos en tres partes: entrenamiento, validaciòn y prueba.

Vamos a dividir el conjunto de datos de manera que

- El 20 % de los datos va a la validaciòn.
- El 20 % va a la prueba.
- El 60 % restante va a entrenamiento.

Esto lo hacemos mediante el siguiente còdigo:

```
set.seed(1234)
trvaltest <- function(dat,prop = c(0.6,0.2,0.2)){
  nrw = nrow(dat)
  trnr = as.integer(nrw *prop[1])
  vlnr = as.integer(nrw*prop[2])
  set.seed(123)
  trni = sample(1:nrow(dat),trnr)
  trndata = dat[trni,]
  rmng = dat[-trni,]
  vlni = sample(1:nrow(rmng),vlnr)
  valdata = rmng[vlni,]
  tstdata = rmng[-vlni,]
  mylist = list("trn" = trndata,"val"= valdata,"tst" = tstdata)
  return(mylist)
}
outdata = trvaltest(df,prop = c(0.6,0.2,0.2))
df_train = outdata$trn; df_val = outdata$val; df_test = outdata$tst
```

```
head(df_train)
```

```
##             make              model year               engine_fuel_type
## 2463       Dodge         Challenger 2016               regular_unleaded
## 2511   Chevrolet          Chevy_Van 1997               regular_unleaded
## 10419 Volkswagen             Tiguan 2016 premium_unleaded_(recommended)
## 8718   Chevrolet        S-10_Blazer 1992               regular_unleaded
## 2986   Chevrolet           Corvette 2015 premium_unleaded_(recommended)
## 1842  Volkswagen Beetle_Convertible 2015 premium_unleaded_(recommended)
##       engine_hp engine_cylinders transmission_type     driven_wheels
## 2463        305                6         AUTOMATIC  rear_wheel_drive
## 2511        200                6         AUTOMATIC  rear_wheel_drive
## 10419       200                4         AUTOMATIC front_wheel_drive
## 8718        160                6            MANUAL  rear_wheel_drive
## 2986        455                8            MANUAL  rear_wheel_drive
## 1842        210                4            MANUAL front_wheel_drive
##       number_of_doors  market_category vehicle_size vehicle_style
## 2463                2      Performance        Large         Coupe
## 2511                3              N/A        Large     Cargo Van
## 10419               4        Crossover      Compact       4dr SUV
## 8718                2              N/A      Compact       2dr SUV
## 2986                2 High-Performance      Compact         Coupe
## 1842                2      Performance      Compact   Convertible
##       highway_mpg city_mpg popularity  msrp
## 2463           30       19       1851 26995
## 2511           17       14       1385  2000
## 10419          26       21        873 28700
## 8718           22       16       1385  2000
## 2986           29       17       1385 59160
## 1842           31       23        873 29895
```

Entrenamiento:

Explique brevemente qué se está realizando en cada linea.

Realice una transformación logaritmica a los datos de entrenamiento, validación y prueba para la varible objetivo, y guardelo en la variables y_train, y_val, y_test, por ejemplo, y_train = log1p(df_train$msrp)

¿Qué se hizo en este caso?

Elimine de df_train, df val, df test, la variable objetivo. ¿Por qué hacemos esto?

```
library(dplyr)
```

```
## Warning: package 'dplyr' was built under R version 3.6.3
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
y_train_orig =  select(df_train, msrp)
y_val_orig = select(df_val, msrp)
y_test_orig = select(df_test, msrp)
```

```
y_train = log1p(y_train_orig)
y_val = log1p(y_val_orig)
y_test = log1p(y_test_orig)
```

```
df_train <- df_train[ ,!colnames(df_train)=="msrp"]
df_val <- df_val[ ,!colnames(df_val)=="msrp"]
df_test <- df_test[ ,!colnames(df_test)=="msrp"]
```